

Lab 4 - Inter-VLAN Routing

Lab 4: Inter-VLAN routing

Lab exercise

Your task is to configure the network such that PC-A in VLAN 2 can ping PC-B in VLAN 3 across the switches. In the topology shown in Figure 13-4, you can always swap the PCs for routers and use the fast Ethernet interfaces to connect to the switches.

Purpose

This topology is known as a router on a stick. When you have VLANs on your network, they must each reside in their own subnet (please refer to the theory notes for more on that). In order for subnets to be able to communicate, you must have a router (higher-level switches can usually perform routing functions, but this is outside the scope of the CCNA).

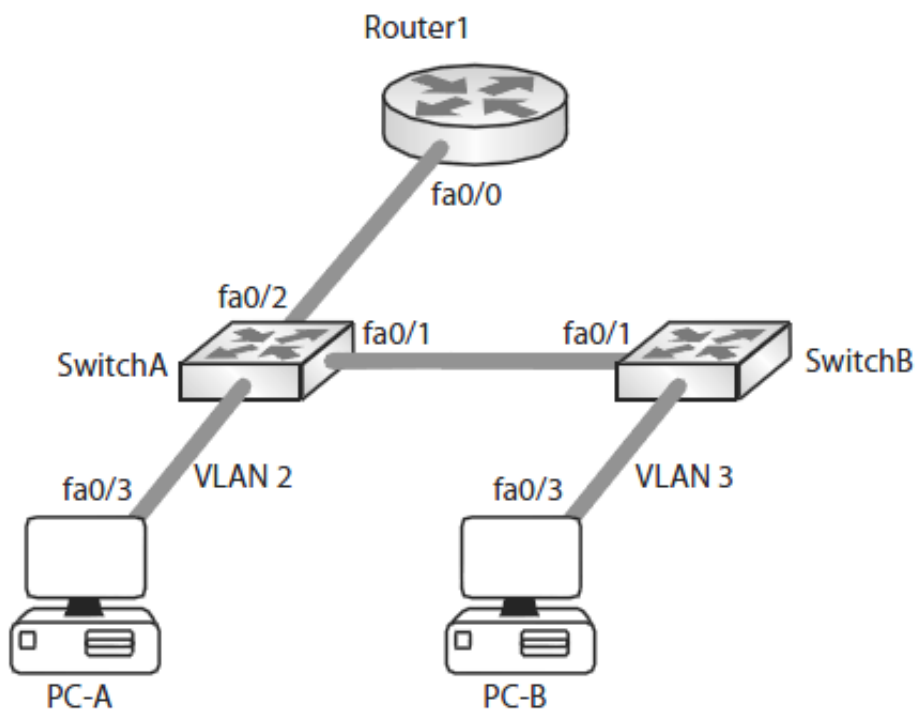


Figure 13-4: Inter VLAN Routing

Vlan 2 – 192.168.2.0/24, PC-A 192.168.2.2/24

Vlan 3 – 192.168.3.0/24, PC-B 192.168.3.2/24

Because you may have only one available fast Ethernet interface on your router, you will need to divide the physical interface into logical sub-interfaces. For example, if your router interface is fast Ethernet 0/0, then the sub-interface would be 0/0.1. It is common practice to name your sub-interface to match your VLANs. So, you would name the sub-interface for VLAN 2--fast Ethernet 0/0.2.

In order to use the router Ethernet interface as a trunk connection to the switches the encapsulation type must be set to dot1q. Depending upon your model of switch, this may be your only encapsulation choice so the encapsulation command may not work (this applies to 2950 model switches).

One last note, in the above lab you may swap PCs for fast Ethernet interfaces on routers. You will need to add static routes on the routers though (see below).

Lab objectives

1. Configure SwitchA to trunk with SwitchB and Router1 using 802.1q
2. Configure SwitchB to trunk to SwitchA using 802.1q
3. Configure port fa0/3 to be in the correct VLANs on both switches
4. Configure Router1's fa0/0 interface with two sub-interfaces in the correct VLANs and with correct IP addresses (.1 in respective subnets)
5. Configure the PCs with respective Gateway addresses (address of the sub-interface on Router1)
6. Ping from PC-A to PC-B.

Lab walk-through

1. To configure SwitchA for trunking on relevant ports. We need trunking ports because they can carry multiple VLAN information. Do the following:

```
Switch#configure terminal
Switch#(config)#hostname SwitchA
SwitchA(config)#interface range fa0/1 - 2
SwitchA(config-if-range)#switchport trunk encapsulation dot1q
SwitchA(config-if-range)#switchport mode trunk
```

Please note--you need a space between the interface numbers if you are using the range command and the command may not work if your IOS version does not support it. If you are using different model switches, your interfaces may show as 1/1, etc. Also, the encapsulation dot1q will not work if you are using a 2950 model switch because it can only use dot1q.

You can check encapsulation type with the “show interface trunk” command.

```
Switch#show interface trunk
```

Port	Mode	Encapsulation	Status	Native vlan
Fa1/2	on	802.1q	trunking	1

2. To configure SwitchB for trunking, do the following:

```
Switch#config t
Switch#(config)#hostname SwitchB
SwitchB(config)#int fa0/1
```

```
SwitchB(config-if)#switchport trunk encapsulation dot1q
```

```
SwitchB(config-if)#switchport mode trunk
```

(Please note—you should not need the encapsulation command on 2950 switches because this is the default).

3. Access ports are used to connect hosts to the switch. They should actually be access ports by default, but it is useful to know the command. To configure the access ports, do the following.

```
SwitchA(config)#interface fa0/3
```

```
SwitchA(config-if)#switchport mode access
```

```
SwitchA(config-if)#switchport access vlan 2
```

```
SwitchB(config)#interface fa0/3
```

```
SwitchB(config-if)#switchport mode access
```

```
SwitchB(config-if)#switchport access vlan 3
```

Please note--we have put the interfaces into the respective VLANs 2 and 3 with the above commands. Also, if VLAN 2 or 3 is not already created on the switch you may see:

```
SwitchB(config-if)#switchport access vlan 2
```

```
% Access VLAN does not exist. Creating vlan 2
```

You can see from earlier labs that we create vlans with the: "Switch(config)#vlan 2" command.

4. To configure the router port, we need to add sub-interfaces and configure dot1q encapsulation so the interface can trunk the VLANs. Do the following:

```
Router#config t
```

```
Router(config)#hostname Router1
```

```
Router1(config)#interface fa0/0
```

```
Router1(config-if)#no shut
```

```
Router1(config-if)#interface fa0/0.2
Router1(config-subif)#encapsulation dot1q 2
Router1(config-subif)#ip address 192.168.2.1 255.255.255.0
Router1(config-subif)#interface fa0/0.3
Router1(config-subif)#encapsulation dot1q 3
Router1(config-subif)#ip address 192.168.3.1 255.255.255.0
```

(Please note—interface fa0/0.2 is used for VLAN 2. We have also added the correct VLAN number after the “dot1q” command. You can see the output the router expects below.)

```
Router1(config-subif)#encap dot1q ?
```

```
<1-4095> IEEE 802.1Q VLAN ID required, range 1 - 0xFFF.
```

5. (Optional) Depending on the Operating System of the PCs, configure their gateway to be 192.168.2.1 and 192.168.3.1, respectively.

In this example, I have used the switching rack at <http://racks.howtonetwork.net> and used two routers (connected to switches A and B) fast ethernet interfaces instead

of PCs. They are only required to prove that the VLANs can communicate. Fast Ethernet

0/0 on RouterA is given an IP address within VLAN 2 and RouterB in VLAN 3. If

you do use routers, please remember to add a static route, e.g., on the Router with IP

```
add 192.168.3.2 add static route 0.0.0.0 0.0.0.0 192.168.3.1.
```

```
Router(config)#hostname RouterA
```

```
RouterA(config)#int fast 0/0
```

```
RouterA(config-if)#ip add 192.168.2.2 255.255.255.0
```

```
RouterA(config-if)#no shut
```

```
RouterA(config-if)# ^ Z
```

```
RouterA#
```

```
Router(config)#hostname RouterB
```

```
RouterB(config)#int fast 0/0
```

```
RouterB(config-if)#ip add 192.168.3.2 255.255.255.0
```

```
RouterB(config-if)#no shut
```

```
RouterB(config-if)# ^ Z
```

```
RouterB#
```

6. Now ping 192.168.3.2 from PC-A (or RouterA if you are using routers):

```
RouterA#ping 192.168.3.2
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 192.168.3.2, timeout is 2 seconds:
```

```
!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

```
Router#
```

(Please note--I do not know which model of switch you will use for this lab, so it may work slightly differently for you. For mine, I had to set both my switches to VTP server, erase the vlan.dat file, and set up a VTP domain between the switches with the commands:

```
SwitchB#vlan database
```

```
SwitchB(vlan)#vtp domain howtonetwork
```

Running Configuration

```
Router1#sh run
```

```
Building configuration...
```

```
--output truncated--
```

```
hostname Router1
```

```
!
```

```
interface FastEthernet0/0
```

```
no ip address
```

```
duplex auto
```

```
speed auto
```

```
!
```

```
interface FastEthernet0/0.2
```

```
encapsulation dot1Q 2
```

```
ip address 192.168.2.1 255.255.255.0
```

```
!
```

```
interface FastEthernet0/0.3
```

```
encapsulation dot1Q 3
```

```
ip address 192.168.3.1 255.255.255.0
```

!

SwitchA#sh run

Building configuration...

--output truncated--

hostname SwitchA

!

interface FastEthernet1/0

!

interface FastEthernet1/1

switchport mode trunk

!

interface FastEthernet1/2

switchport mode trunk

!

interface FastEthernet1/3

switchport mode access

switchport access vlan 2

!

SwitchB#sh run

Building configuration...

--output truncated--

hostname SwitchB

!

interface FastEthernet1/0

!

interface FastEthernet1/1

switchport mode trunk

!

interface FastEthernet1/2

!

interface FastEthernet1/3

switchport mode access

switchport access vlan 3

!



© 2006-2011 HowtoNetwork.net All Rights Reserved. Reproduction without permission prohibited.